MySQL Incremental Backup

Brief:
Incremental Backup is one of important requirements on production databases with big size. Without a safe incremental backup, you can not tell yourself that you have a reliable production database. Because you must have enough data in order to recover your database in emergency cases. After some search on Internet, I could not find any tool that can do a complete incremental backup for MyISAM and InnodB in a mix environment that applications use both of engines simultaneously (maybe I am not an expert searcher on Google and Internet). So I decided to write the one, but to avoid wasting time and benefit from other open-source solutions, I preferred to add this feature to “automysqlbackup” script that is the best script for full backup in simplicity and widespread use.

mechanism:
- We use Post and Pre feature of automysqlbackup to do an incremental backup. before starting full backup, mysql-backup-pre executes a query to lock whole databases during backup process, because we must freeze the binlog to avoid any change while backup is running. The binlog name and position must not be changed. The binary log and position is very crucial in subsequent incremental backup and will be used as a start point to begin next incremental backups. After finishing full backup, mysql-backup-post kill the lock query.

Lock Query: FLUSH TABLES WITH READ LOCK; SELECT SLEEP(86400)
Find Lock Queries: mysql -u[username] -p[pass] -e "show processlist" | grep "SELECT SLEEP(86400)" | awk '{print $1}'

I strongly advise to setup this backup solution on Slave server not Master to avoid any effect on performance of main database.

Requirement:
- root privileges to install package and update mysql.conf
- mysql-community-client package
- installation automysqlbackup and mysql-incremental

Installation:

- install mysql-community-client package for your distro
  - after installation you must have 'mysqlshow' command.

- install automysqlbackup :
  - download the package from https://sourceforge.net/projects/automysqlbackup/
  - tar -xzf [PathYouSavedTarFile] -C /tmp/
  - cd /tmp/
  - ./install.sh :
    during installation of automysqlbackup, you will be asked about path of automysqlbackup.conf and its binary, you can leave defaults without any change.
  - rm /etc/automysqlbackup/myserver.conf

- Install mysql-incremental:
  - download the package from https://sourceforge.net/projects/
  - tar -xzf [PathYouSavedTarFile] -C /tmp/
  - cd /tmp/
  - cp mysql-incremental to /etc/automysqlbackup/
  - chmod 755 /etc/automysqlbackup/mysql-incremental
  - cp mysql-backup-post /etc/automysqlbackup/
  - chmod 755 /etc/automysqlbackup/mysql-backup-post
  - cp mysql-backup-pre /etc/automysqlbackup/
  - chmod 755 /etc/automysqlbackup/mysql-backup-pre

- update the automysqlbackup.conf:

  Find below parameters, uncomment and change them.

  ```
  CONFIG_mysql_dump_username='Mysql user name. It must has privileges to get Lock'
  CONFIG_mysql_dump_password='Password'
  CONFIG_backup_dir='The backup directory you want to store full and incremental backup'
  CONFIG_db_names=('databaseName1' 'databaseName2' )
  CONFIG_db_month_names=('databaseName1' 'databaseName2' )
  CONFIG_mysql_dump_master_data=2
  CONFIG_prebackup="/etc/automysqlbackup/mysql-backup-pre"
  CONFIG_postbackup="/etc/automysqlbackup/mysql-backup-post"
  ```

- Update my.conf :

  - Binlog format:

  due to some limitation on STATEMENT format, my recommendation is to set ROW based format. For more information please see the 'troubleshoot' section on this howto. You can check the type of binary log format by executing “*select @@binlog_format;*” query. To modify logbin format , you must add “binlog_format   = ROW” to mysql.conf or my.cnf .

  - binlog_do_db:

  You must specify the databases you intend to have the related changes in binary log. Please note if you do not specify any database, any change on any database will be logged into binary log. In this case, if you chose STATEMENT format, maybe you have some trouble when restoring from incremental backup and binlog files. You can add databases to this option:

  ```
  binlog_do_db = DATABASENAME1
  binlog_do_db = DATABASENAME2
  ```

  - expire_logs_days:

  To have binary log files for more time, you can increase this parameters to higher value. My recommendation is 60 days:

  ```
  expire_logs_days        = 60
  ```

  - log-bin :

The directory that binary logs will be stored. In old version of MySQL, mysql-incremenetal could not able to find the correct path. So if you get error about this after executing mysql-incremental, you must update mysql-incremental script to the binary logs path.
- log_slave_updates :
If you are setting up mysql-incremental backup on Slave, you must enable this option.
Normally, a slave does not log to its own binary log any updates that are received from a master server. This option tells the slave to log the updates performed by its SQL thread to its own binary log.
http://dev.mysql.com/doc/refman/5.1/en/replication-options-slave.html#option_mysqld_log-slave-updates

- run automysqlbackup manually to have at least one full backup from your specified databases. After executing successfully, check the /[BackupDirInAutomysqlbackup]/status/backup_info file for added new information about daily backup and for any error info check "/var/log/Backup_Post_Pre_log". The backup file would be under /[BackupDirInAutomysqlbackup]/daily/[DatabaseName]/ .
- Run mysql-incremental manually to have at least one hourly backup. After executing, in error cases, the error info are logged into "/var/log/Backup_Incremental_Log" . The incremental backup files would be under /[BackupDirInAutomysqlbackup]/IncrementalBackup/ .

- Crontab:
you can schedule mysql-incremental for more than one hour. You can find the total time of full backup from backup_status and then based on that value you set an accurate schedule time. Of course mysql-incremental backup does have a mechanism to find any running full backup before start, so there is no concern about conflict between incremental and full backup.

Restore Database:

- In order to restore up to a specific time (point in time recovery), first you must restore one full daily backup and then restore sequentially related incremental backup files.
- Sample:
In sample scenario we intend to recover our data up to 2015-5-01 at 2 AM.
we have set /backup as our main backup dir and testDB as our target database.

1- mysql -u root -p DatabaseName < /backup/daily/testDB/(to be completed when I could connect to real database)
2- mysql -u root -p DatabaseNAme < /backup/IncrementalBackup/2015-5-01_Incremental/testDB/testDB_IncrementalBackup_2015-5-01_00h25m.1
3- mysql -u root -p DatabaseNAme < /backup/IncrementalBackup/2015-5-01_Incremental/testDB/testDB_IncrementalBackup_2015-5-01_01h25m.2
4- mysql -u root -p DatabaseNAme < /backup/IncrementalBackup/2015-5-01_Incremental/testDB/testDB_IncrementalBackup_2015-5-01_02h25m.3

Important notes and Troubleshoot:

- MySQL supports different format for binary logs. Some of Mysql versions use 'statement-based' as binlog format that this type of binlog does have some limitations that we must pay close attention to it when we intent to use it in incremental backup procedure. When mysql is set to statement-base format, it does not able to filter correctly based on databases. if you set 'USE or \u' to change database and then update another database which is not included in binlog-do-db, the statement will be logged in binlog file that it is not desirable state! and will expose some issue when restoring based on specific database and also if you change to another database that is not included in binlog-do-db, and update a database which is included in binlog-do-db, the statement will not logged to binlog file. our purpose from adding databases to binlog-do-db is to filter based on database,but it does not work as expected. If “USE or \u” is not executed before running queries, ,mysqlbinlog can not extract 'update queries' related to one database. We will explain more this issue with below scenarioes:

```
 databases:
 - binlog
     - person (table)
  - binlog2
     - person (table)
```

binlog-do-db=binlog2 (it is supposed only change of this database are logged to binlog file)
--------Scenario 1---------
\u binlog2|
insert into person (data) values ('17') ---> loged in binlog *desired state*
insert into binlog.person (data) values ('25'); ---> logged in binlog (target database is 'binlog') *undesired state*
--------Scenario 2---------
\u binlog
insert into person (data) values ('17') ---> is not logged in binlog *desired state*
insert into binlog2.person (data) values ('25'); ---> is not logged in binlog (target database is 'binlog2') *undesired state* because the binlog2 database is begin changed, so we want to have this change,but it will not logged in logbin file
--------Scenario 3---------
if you just connect to database without any “USE or \u” statement, all of updates on any databases will be logged, but mysqlbinlog can not able to filter based on specific database, so that is not desirable state for our purpose in incremental backup. Using “USE or \u” before executing update queries, is very important. Because mysqlbinlog finds update queries based on USE statement in binlog file.

- I did try to log everything in log file so you can find enough information in log “/var/log/Backup_Post_Pre_log” and “/var/log/Backup_Incremental_Log”
- file /[SpecifiedBackupDirInAutomysqlbackup.conf]/status/backup_info contains the detail info about when the backup is finished(Unix Time format),the binlog name and position that until there the backup has been done successfully, type of backup, number of backup from previously full backup and duration of backup.

Sample backup_info:

```
1431043501,mysql-bin.000026,120,Daily,2015-05-08,0,24
1431044701,mysql-bin.000026,120,Hourly,2015-05-08,1,1
```

here are description of different value:
1th) 1431043501 : indicates the time when the backup has been finished. You can run “ date --date @1431043501“ command on the server the backup has been done to view it in human readable format.
2th) Mysql-bin.000026 : indicates the binary log name that backup up to this file has been done.
3th) 120 : indicates the position of binlog that backup up to this position in binary log has been done.
4th) Daily/Hourly: indicates type of backup. Daily does mean the full backup by automysqlbackup script and Hourly is done by mysql-incremental script.
5th) 2015-05-08: The date that backup has been done. This date will be used in creating directory for incremental backup and also as a base for restore hourly backups. In restoring procedure, first a full backup is restored and then sequentially other incremental backup are restored.
6th) 0 : indicates number of backups from previous full backup. 0 does mean the backup is full and others mean hourly. This number is very important in restoring procedure.
7th) 24: The backup duration in second.